Experimental Validation of an Unpowered Unmanned Aerial System: Application to Forced Landing Scenarios

Luis Mejias, Pillar Eng

Abstract— **The ability to perform autonomous emergency (forced) landings is one of the key technology enablers identified for UAS. This paper presents the flight test results of forced landings involving a UAS, in a controlled environment, and which was conducted to ascertain the performances of previously developed (and published) path planning and guidance algorithms. These novel 3-D nonlinear algorithms have been designed to control the vehicle in both the lateral and longitudinal planes of motion. These algorithms have hitherto been verified in simulation. A modified Boomerang 60 RC aircraft is used as the flight test platform, with associated onboard and ground support equipment sourced Off-the-Shelf or developed in-house at the Australian Research Centre for Aerospace Automation (ARCAA). HITL simulations were conducted prior to the flight tests and displayed good landing performance, however, due to certain identified interfacing errors, the flight results differed from that obtained in simulation. This paper details the lessons learnt and presents a plausible solution for the way forward.**

I. Introduction

While UAS technology has been proven in the military area, their benefits for civilian applications are yet still to be seen. Regulatory entities around the world are currently defining and establishing the minimum requirements and guidelines for the use of Unmanned Aerial Systems (UAS) in a civilian context. Without a clear regulatory framework this industry will not be able to develop in full capability. Since many of the proposed missions for civilian UASs will involve flying over populated areas and in airspace occupied by manned aircraft, policy makers are conscious of the repercussions that a major UAS accident could have on public acceptance of this technology. This may be arguably the main factor which has prevented these UAS trials from becoming full-scale commercial operations, as well as restricted operations of civilian UASs to only within segregated airspace. As a consequence much research is underway in developing technologies to enhance UAS autonomy. In particular, two of the most important technology enablers for UAS identified by regulators and governments are, the capability to see and avoid, and the capability to perform autonomous forced landings [1], [2]. A forced landing is an unscheduled event in flight requiring an emergency landing, and is most commonly attributed to engine failure, failure of avionics or adverse weather.

State-of-the-art automated navigation systems already exist for UAS, however there is a lack of automation in scenarios where the aircraft experience an emergency situation. To date, the most commonly employed method to allay the severity of a UAS forced landing is the use of parachutes or parafoils to retard the rate of descent, while still providing some degree of controllability for the aircraft [3]. Whilst this concept is attractive in that it still enables limited vehicle controllability even when both the engine and control surfaces have failed, it is highly susceptible to wind gusts and other atmospheric effects which may adversely affect the final impact point. Our approach is based on the premise that the UAS have still some degree of flight control so that the aircraft is able to manoeuvre to a desired landing site.

To date, the only reported successful UAS forced landing involves the U.S. Air Force Global Hawk, which performed a gliding descent under remotely-piloted control (RPC) to an emergency airstrip in 2006 [4].

This paper presents the flight test outcomes of a forced landing system for UAS, which have

Luis Mejias is with the Australian Research Center for Aerospace Automation at Queensland University of Technology, Brisbane, Australia. luis.mejias@qut.edu.au

Pillar Eng is with the Technology Solutions Group at QinetiQ Pty Ltd, Adelaide, Australia peng@qinetiq.com.au

provided preliminary feedback on the performance of the planning and guidance algorithms in a real-world context. The lessons learnt from these results will serve to enhance future work in this area.

This paper is structured as follows. Section II introduces the basic approach to guidance, planning and control of the UAS. Section III describes the way the experimental flight tests were conducted. Section IV outlines the hardware and software setup for the experiments. Section V presents the outcomes of the flight test. Finally, section VI describes some of the lessons learnt and future work planned.

II. Path Planning, Guidance and Control

The concept behind the path planning approach was initially influenced by actual piloted forced landing procedures and patterns outlined in [5]. These procedures generate the initial target waypoints forming a cone, which defines the airspace to which the unpowered aircraft can fly in nil wind conditions. Algorithms based on these procedures are designed to maximize the probability of the aircraft reaching the desired landing site, yet their complexity, and the fact that they use fixed distances between waypoints, as well as rely on a fixed airspeed may result in their undoing.

For instance, some of these procedures may require the aircraft to fly back and forth between two waypoints, which can bleed off too much altitude before the final turn for the aimpoint. In some other cases, instability may result from the aircraft constantly turning while seeking for a point below the projected glide slope. Hence, it may be more advantageous to design a reusable algorithm which simply uses a given starting and goal location to construct the required path, while taking into account the aircraft dynamics.

From the literature, it has been found that such a path can best be described by trajectories derived from Dubins curves [6]. Dubins curves allows the construction of optimal planar paths to move a vehicle (such as a car or aircraft flying at constant altitude) from an initial to a goal location defined in terms of position and heading. These Dubins-path-based approaches have low computational burden and simple design procedure. However, they are constrained to 2-D applications. Two examples where Dubins curves were used in 2D [7] and 3D [8] UAS path generation were proposed by Kim et al., and Babaei & Mortazav, respectively.

In this work, a new 3-D Dubins path planning algorithm [9] was developed. This algorithm shares some similarities with that described by [10], however, the basic idea presented there has been greatly extended into a novel planning approach for the gliding descent of fixed-wing aircraft. This algorithm is fully disclosed in [9], [11], and the results of testing using this algorithm are presented in this paper. For testing in simulation, a 6 degree-of-freedom model of a Boomerang 60-size UAS was adopted as it represents the aircraft to be used for flight tests. Note that both path planning algorithms disclosed here are not restricted to any specific aircraft type, but can be applied to both manned and unmanned fixed-wing aircraft of any size and type.

For guiding the aircraft, a 3-D nonlinear algorithm has been designed to control the vehicle in both the lateral and longitudinal planes of motion. The lateral guidance approach is based on the work presented in [12]. However, this algorithm has been enhanced to include wind information in the guidance logic, rather than merely treating wind as an adaptive element for the control system. This addition has demonstrated robust, linear path following in strong winds. Secondly, by making a simple assumption in formulating the equation for following a circular path, the guidance logic has been simplified without sacrificing performance.

In addition, a longitudinal guidance and control element has been implemented that caters for the dynamics of powerless flight. Two different approaches were trialled in this regard, the first uses PID gain scheduling to control the aircraft pitch angle, and the second uses the centripetal acceleration between the aircraft and flightpath to calculate the desired pitch angle commands. It is revealed that the second approach outperforms the first due to its robustness [9], however, only the results of using the second approach is presented in this paper.

Finally, following well-established aircraft control design procedures [13], the design has been separated into two modes: an inner control loop that provides aircraft dynamic stability, and an

outer guidance loop that generates the required acceleration and position commands to follow a path. Details concerining the lateral guidance algorithm, named the Enhanced Nonlinear Guidance (ENG) algorithm, as well as the two longitudinal guidance algorithms, named the Flight Path Following Guidance (FPFG) and Modified Proportional Navigation (MPN) algorithms respectively, can be found in [9], [11].

This paper is devoted to the experimental design, test and validation of our approach that has been tested in simulation previously as disclosed in [11], [14], [15], [16]. Therefore, the contribution derived from this paper is in the experimental procedure and field report under realistic conditions that demonstrate the technology readiness level of this approach towards its maturity.

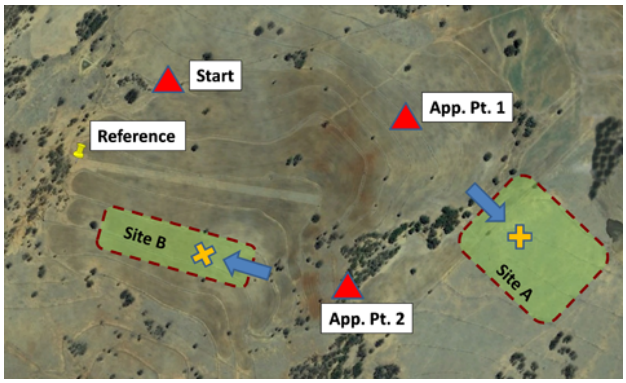

Fig. 1. Aerial view of the Burrandowan test site. The two candidate landing sites are labeled as Site A and Site B, and have their preferred direction of approach indicated by the blue arrows. Shown also is the reference point for translating between diferent coordinate systems, the approach point and aimpoint for each site, as well as the starting waypoint for all forced landing descents

III. Experiment Design

In order to verify that the forced landing path planning and guidance algorithms will function as intended, a specific test scenario has been designed that assesses every aspect of the 3-D Dubins curves planning algorithm, as well as the ability of the ENG and MPN guidance algorithms to follow the prescribed path. The implementation of this test scenario is described below.

The forced landing flight tests are conducted at a remote airstrip located at Burrandowan, in the state of Queensland, Australia. For these series of flight tests, the aimpoints of the forced landing sites are assumed to be already calculated by a higher level multi-criteria decision maker, and are located one-third of the way into the landing sites, labeled as Site A and Site B in Figure 1.

The preferred directions of approach for landing are also assumed to have been precalculated by the same decision maker, and are indicated by the blue arrows at the start of each site. Guidance for calculating the aimpoints and approach directions are given in [5], and are used for the flight tests described in this paper. From the aimpoints, the approach points can be calculated, and these are shown by the red triangles in the figure. The approach points are the final waypoints to which the aircraft will be guided in flight tests. Although the airstrip located between the selected sites (demarked by a yellow push pin) is also suitable for use as a forced landing site, it is not used in these tests. The yellow push pin also marks the location of the reference point for translating from the Earth-Centred-Earth-Fixed (ECEF) to East-North-Up (ENU) Cartesian coordinate system, as required by the guidance algorithms. The relationship between the aircraft and the ECEF and ENU coordinate systems are illustrated in Figure 2.

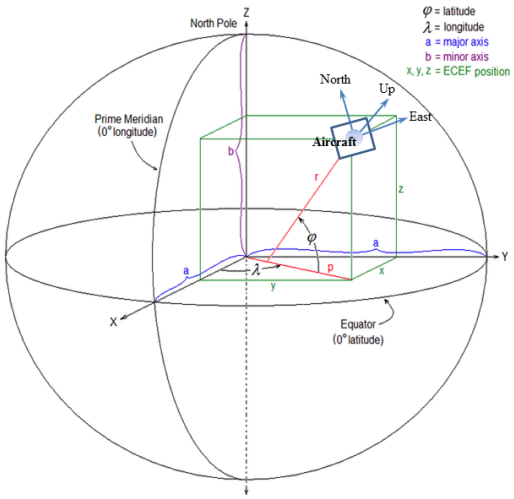

Fig. 2. Relationship between ECEF (earth-centred) and ENU (body-centred) coordinate systems.

The flight tests are conducted in the following manner. First, a pilot flies the UAS under radio control from takeoff to approximately 800 ft and facing the general direction of the starting point

for a forced landing. Control is then translated to the Micropilot® HORIZON^{mp} ground station software [17], which will guide the aircraft the remainder of the way to the starting point (approx. at 1500 ft). Once the aircraft comes within 50 m of that point, the ground station software will reduce the throttle setting to idle, and a ground operator will then start the onboard flight computer containing the path planning and guidance code using a point-to-point radio modem link.

Since the operational ceiling has been limited by CASA [1] 1500 ft, the aircraft will initially conduct only one spiral to lose altitude, before heading for the approach point at Site A, located at an altitude of 460 ft. While enroute to Site A, a simulated low altitude condition will cause the UAS to head for Site B, which in this test assumes the role of a more feasible site. The approach point at Site B has an altitude of 100 ft. However, due to the differences in altitude between the starting position and the approach points, a joining contour is required to link the spiral path to the 3-D Dubins curve. Thus, not only will this test assess the ability of the path planner to construct a feasible path that also accounts for the vehicle dynamics, but will further test its replanning capability. In addition, the nature of the path shapes will also challenge the ability of the guidance algorithms to follow those paths.

A schematic diagram illustrating the test procedure described above is depicted in Figure 3.

Prior to conducting the actual flight tests, the forced landing scenario is run inside the HORIZON^{mp} simulator. This software program is capable of simulating a variety of real-world conditions, such as avionics and GPS failures, loss of flight control and winds. It is also able to accept user-configurable flight plans, written as a *.fly* file. For simulation and for the actual flight tests, a *.fly* file is written that autonomously guides the UAV to the starting location and to be at the correct altitude and heading prior to the commencement of each test. Once the test is started, control is transferred to the path planning and guidance algorithms. The planner generates (in real-time) a list of waypoints to the approach point. This information is then passed to the guidance algorithm which analyses

[1]Civil Aviation Safety Authority in Australia

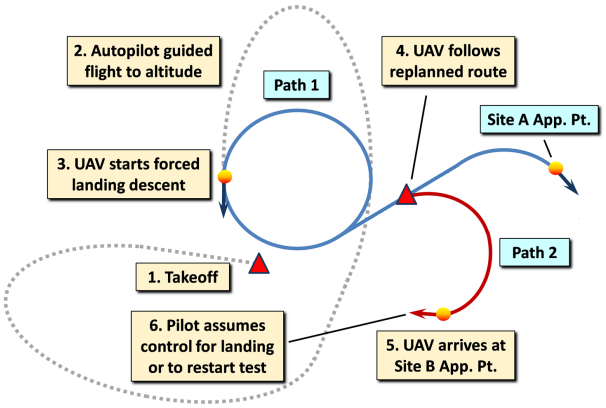

Fig. 3. Schematic diagram of the procedures followed in conducting the UAV forced landing flight tests. The dotted line represents the path flown by the UAV to the starting location of the forced landing descent. The blue line represents the forced landing flight path to Site A, and the red line represents a change to the path while the UAV is in flight, and which the UAV must follow to arrive at Site B.

the aircraft position in relation to the planned path, and outputs a series of roll and pitch commands to the onboard autopilot. The autopilot in turn commands the necessary servo deflections to steer the aircraft.

IV. Experimental Setup

A. Testbed Platform

The testbed used is the Boomerang 60 model aircraft from Phoenix Models, with a wingspan of 2.1 m and measuring 1.5 m from nose to tail. This model is powered by an O.S. 90 FX engine, and has been modified from a high to low-wing configuration for added manoeuvrability. In addition, the wingspan has been increased from 1.45 m to 1.9 m to support a total take-off mass of 8 kg, including all onboard avionics and a full tank of fuel. Figure 4 presents a schematic diagram of the unmanned aircraft system.

The heart of the onboard electronics is comprised of an off-the-shelf MicroPilot MP2128g autopilot and an external PC/104 CPU, used as the flight computer. The flight computer communicates with the autopilot via an RS232 serial communications link, to receive aircraft states and send desired commands. The autopilot is connected to the elevator, aileron, rudder and throttle servos via an external servo control board, and transmits telemetry data via a radio frequency (RF) modem

(RF Modem 1) to the Mobile Operations Centre (MOC). The two two-way communication links used are RF Modem 1 with RF Modem 2, and RF Modem 3 with RF Modem 4.

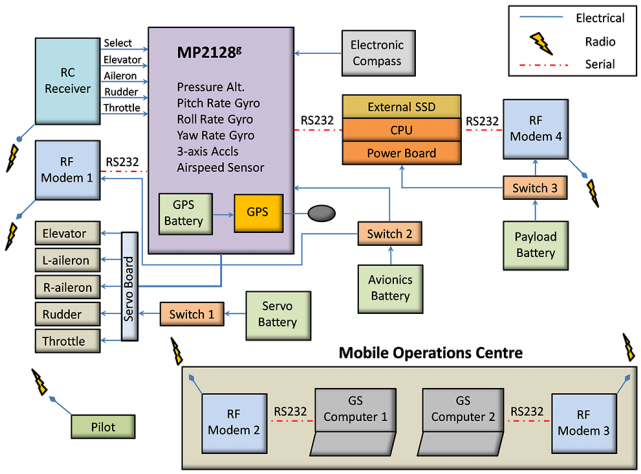

Fig. 4. Boomerang-60 avionics architecture

Fig. 5. Flight activities. Clockwise from top: The Boomerang-60 UAS in flight; the ground operator, who communicates with the pilot via UHF radio and constantly monitors the progress of the flight; the pilot, who is on standby for takeoff and landing, as well as any emergencies, and finally, the MOC which houses the ground operator and associated computing and communications equipment.

B. Hardware Details

1) Flight Computer: The flight computer consists of a LiPPERT Cool LiteRunner 2 PC/104 embedded CPU hosting an AMD Geode LX800 processor running at 333 MHz and with 256 Mb of RAM. The Cool LiteRunner uses passive cooling and provides for analog VGA output, RS232/RS485/RS422 serial data communications, Ethernet, PS/2 keyboard and mouse connections, as well as conventional PCI and IDE expansion slots. An additional 8 GB Transcend 2.5" IDE Solid State Drive (SSD) is used to host the Debian 5.2 operating system, as well as the path planning and guidance software.

2) Autopilot: The MicroPilot MP2128g autopilot is one of the smallest autopilots on the market today, measuring 10x4x1.5 cm and weighing only 26 g, it contains a full avionics suite including GPS, 3-axis gyroscopes and accelerometers, a pressure altimeter, pressure airspeed sensor and an electronic compass. This autopilot uses PID gain scheduling for control stability, as well as a rudder-aileron feed forward gain for improved turning performance.

The autopilot supports both computer-in-control (CIC) and pilot-in-control (PIC) modes, and comes with the HORIZON^{mp} ground control software, which allows an operator to receive telemetry data as well as send telecommands to the aircraft. An additional plug-in capacity allows code written by the user to run alongside the autopilot, and even to modify certain settings of the autopilot.

3) Communications: A Spektrum DX7 7-Channel, 2.4 GHz spread spectrum RC system is used to fly the aircraft in PIC mode. The DX7 transmitter is capable of storing memory for up to twenty different models, and was chosen due to its robustness to noise. The 2.4 GHz spread spectrum system uses the Spektrum AR7000 receiver with dual-linked satellite antennas, and supports an operational range of up to 2 km.

Two MicroHard 900 MHz wireless radio modems (operating in licensed band) are used to communicate between the aircraft and ground operators. This industrial grade radio modem provide 19.2 kbps throughput of data and support Point-to-Point communications. The radio modems are used by the autopilot and the flight computer for telemetry purposes.

Figure 6 depicts the mounting positions of the

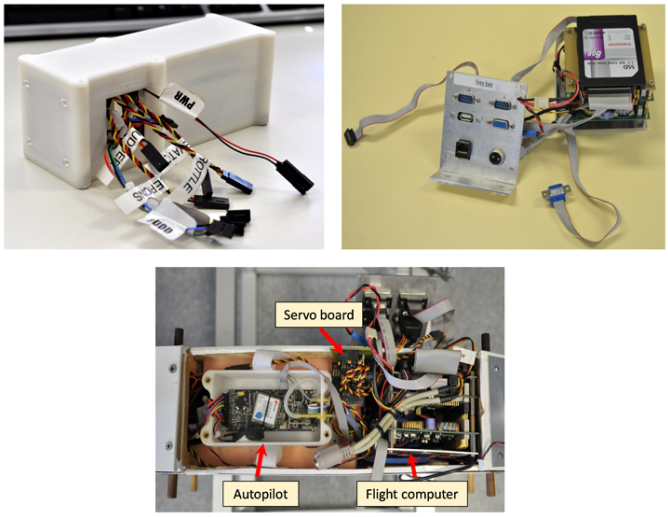

Fig. 6. Clockwise from top: The MicroPilot autopilot box; the flight computer stack and interface plate; location of avionics inside the fuselage.

avionics inside the fuselage

V. Discussion of Results

Prior to conducting the flight tests, the scenario for the experiment is run inside the HORIZON^{mp} simulator. We use the same configuration files for simulation and for the actual flight test. The configuration files allow us to define starting location, initial altitude and heading prior to the commencement of each test.

Figure 7 illustrates the UAS flightpath when following the planned route in nil wind conditions. The aircraft starts the forced landing descent at the position indicated and completes approximately one-and-a-half spirals before joining the standard Dubins path. At Point B, the path is replanned and the aircraft follows the new route to arrive at the approach point. The lateral and vertical miss distances at the approach point are 10 m and 20 m respectively. Considering that the GPS receiver modelled has a nominal error of 10 m, these associated errors are very reasonable. In addition, the average track error is approximately 30 m, which is within the specified bounds, with a maximum deviation of approximately 30 m laterally and 200 ft vertically at Point A. The roll and pitch performances for the flight is depicted in Figure 8, and show good tracking of the input commands with desirable control responses. Note that the constant pitch angle command of 10 degrees is a function of the guidance algorithm, which seeks to direct the aircaft to best attain the approach point in the ambient atmospheric conditions (nil wind in this case).

A total of nine flights were completed in two days, and Figure 5 depicts the various activities carried out as part of the forced landing flight test. From the flights, two specific cases have been chosen to illustrate the performances of the path planning and guidance software. The first case is shown in Figure 9. Here, the UAV starts initially at a point slightly away from the planned forced landing start point, and facing in a direction different to that intended (due South). This is seen in Figure 9a, and is primarily due to the placement of the waypoints that are used to guide the UAS to the start point. These waypoints are chosen in the HORIZON^{mp} program, and are later amended as shown in the next example. The UAS then attempts to follow the descent flight path indicated by the black line, with the incident wind vectors (calculated by the autopilot) depicted by the green arrows. The maximum wind speed encountered was 4.6 m/s, with an average of 2.4 m/s, which is far less than the aircraft airspeed of greater than 20 m/s. However, as seen in Figure 9a, the UAS strays too far from the intended path while circling to lose altitude before heading for the approach point at Point C. When the aircraft is at Point A, the ground operator detects that the aircraft has passed below a previously agreed minimum safe altitude. He then transfers control to the pilot who then brings the aircraft in to land. Due to the large miss distance between the aircraft and the intended path, the UAS never arrives at the intended approach point (Point D), which would require path replanning to be conducted. The red line, indicating the replanned path, has been included merely for reference.

An oblique view of the descent is shown in Figure 9b, and it can be seen that vertical tracking is quite poor. A post flight analysis of the recorded flight data revealed that the GPS altitude used to calculate the pitch required was only available at 4 Hz, and this slow update rate could not have matched the rate the roll commands were received (30 Hz). This resulted in the aircraft receiving pitch

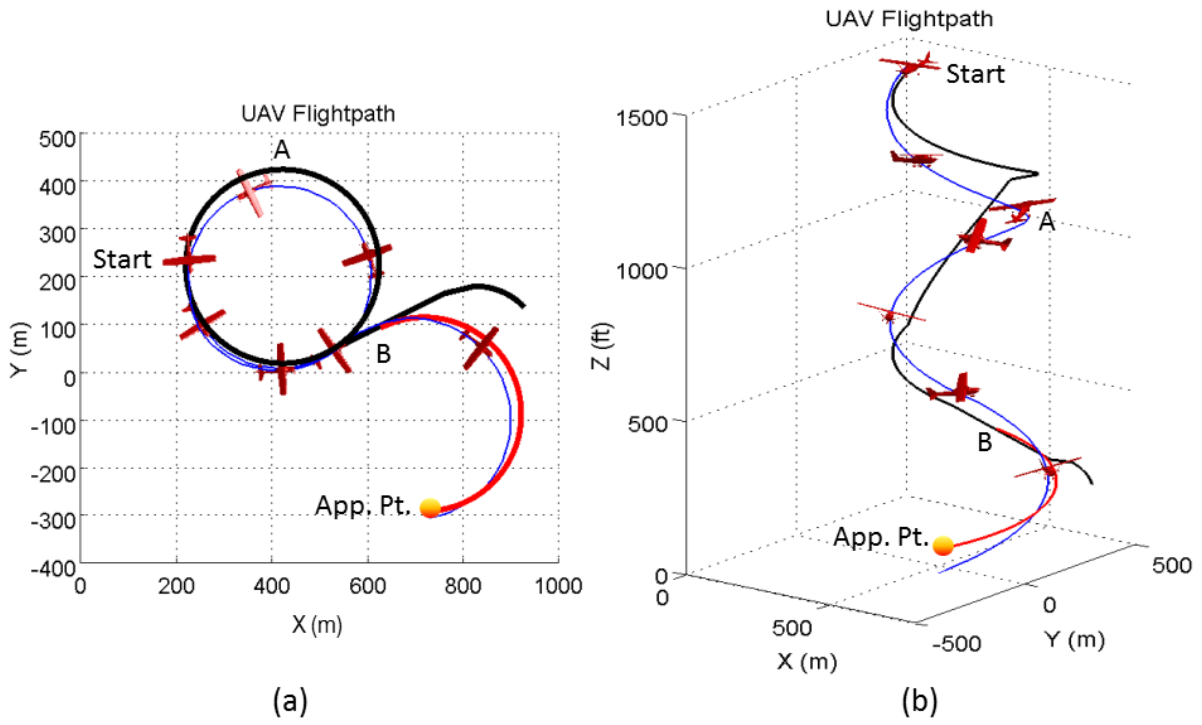

Fig. 7. Path replanning and tracking in nil winds using the HORIZON^{mp} Simulator, showing (a) Top view of the aircraft response, and (b) An oblique view of the same.

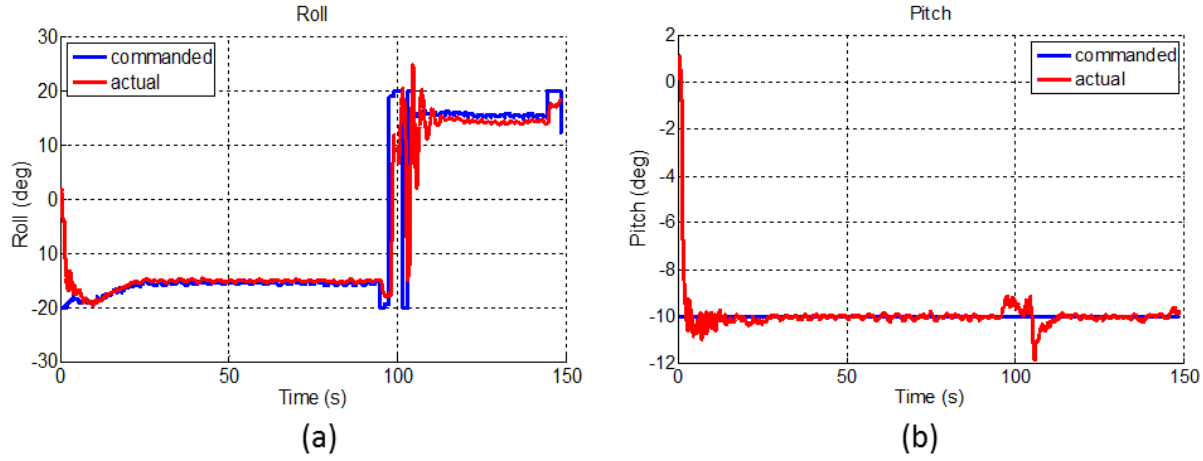

Fig. 8. Aircraft control response from a simulated forced landing descent in the HORIZON^{mp} Simulator, showing good tracking of the input commands in (a) Roll and (b) Pitch.

commands that were greatly out-of-phase with the roll commands, and having to ”chase” the effects of the roll commands at the current aircraft position. This error was corrected in subsequent flights by using the barometric altitude, updated at 30 Hz. Considering a log of the pitch values plotted in Figure 9d, it would also seem that the commanded pitch assumed the form of a hysteresis controller. This can be explained by the lookup table used to convert from desired airspeed to desired pitch in the MPN algorithm (Section II), where an airspeed of greater than 24.5 m/s translates to a pitch angle

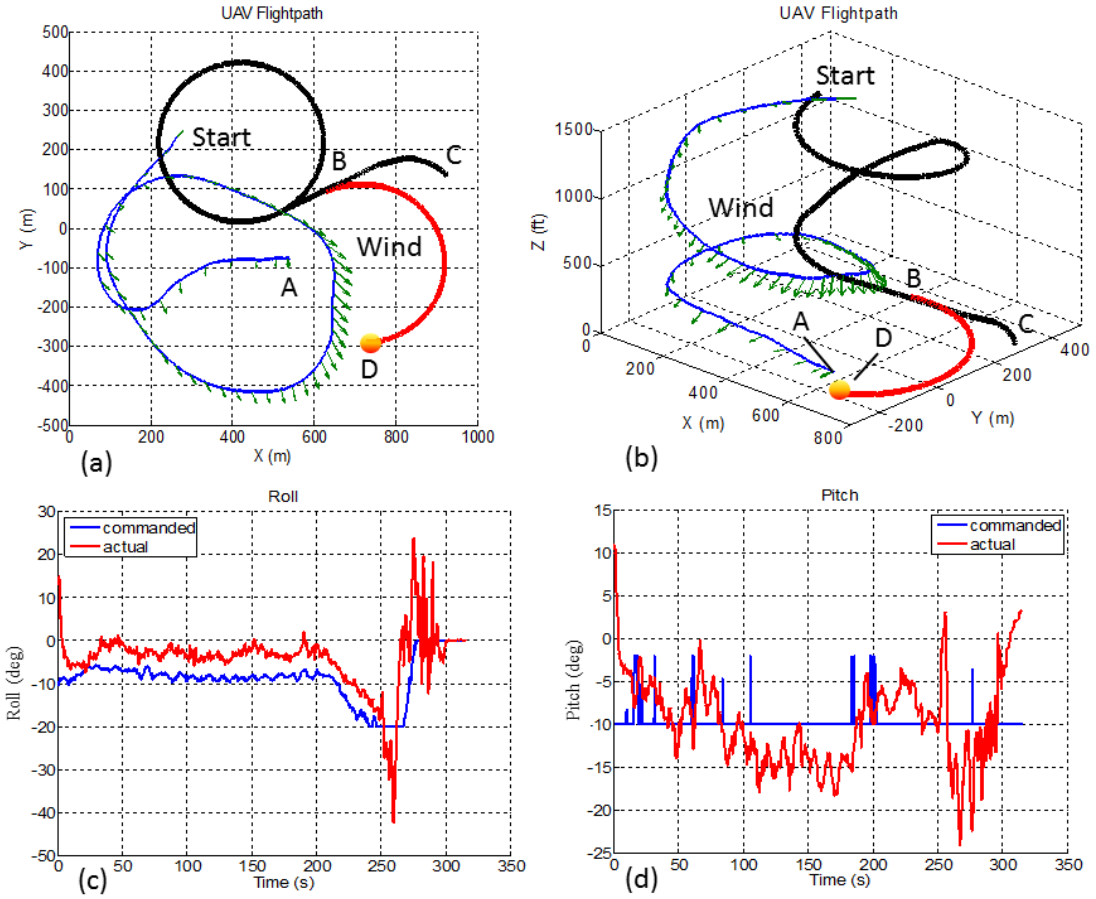

Fig. 9. Results from flight test example 1. (a) Top view of flight path; (b) Oblique view of the same; (c) Roll performance; (d) Pitch performance

of -10 degrees, and -14 degrees is the maximum allowable pitch angle (for safety). The lookup table (Table I) used is:

Airspeed (m/s)	Pitch (deg)
16.89	-2
20.86	-6
24.45	-10
28	-14

TABLE I

LOOKUP TABLE USED TO CONVERT DESIRED SPEED TO DESIRED PITCH ANGLE. SEE [9], PAGE 78 FOR MORE DETAILS

The value of -10 degrees was chosen in this experiment (based on simulated results) as the maximum permissible pitch command to afford a measure of safety and guard the aircraft from diving too steeply. However, this consideration was later shown to be too conservative, as subsequent flights revealed that at -10 degrees the aircraft was travelling too slowly, almost at the stall speed. It can also be seen from Figure 9d that the autopilot pitch controller did not have enough authority to follow the commanded pitch, or the aircraft had a rate of pitch that prevented it from achieving tight following of the pitch commands, even though Figure 8 has indicated otherwise. This can be explained by the fact that the autopilot PID gains were previously tuned for an aircraft weight of 7.4 kg, as opposed to the actual takeoff mass of 7.8 kg.

Considering the roll performance in Figures 9c, it can be seen that the actual roll follows the commanded roll quite well, albeit with an offset; this could be simply due to air turbulence affecting the longitudinal motion of a light aircraft. However, when comparing the commanded roll with

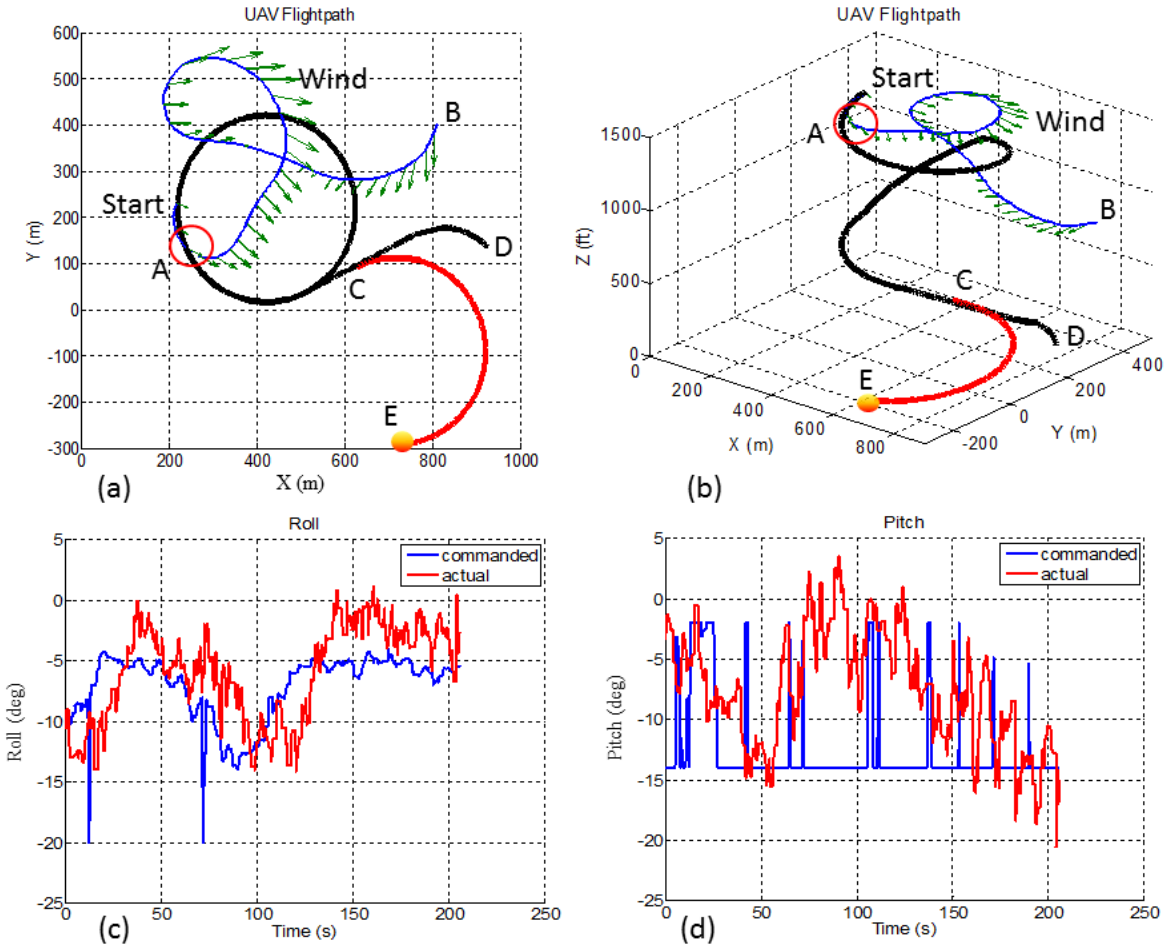

Fig. 10. Results from flight test example 2. (a) Top view of flight path; (b) Oblique view of the same; (c) Roll performance; (d) Pitch performance

that obtained in simulation, it was observed that the former is much less than that required for tight path following. Since the airspeed is used in calculating the required lateral acceleration and hence the required roll, the low airspeed attained (due to the limitations imposed on the pitch angle) could most certainly have contributed to this error.

The momentary large peaks in the actual roll and pitch angles towards the end are caused by the switch from CIC to PIC mode, when the pilot resumed control of the aircraft. Finally, the fact that the normal modes of roll and pitch control as used in the autopilot were circumvented to accept inputs from the guidance software, may have also contributed to the noisy signals received, such as the spikes in the commanded pitch, as it is possible to have bypassed any internal noise filters in the process.

The second case is depicted in Figure 10, and shows the UAS initially starting at the correct location and with the correct heading. The wind velocities are indicated by the green arrows, with a maximum wind speed of 3 m/s and and average wind speed of 2.2 m/s. In this test, the barometric altitude was used instead of the GPS altitude, as it is more accurate for low altitude flights. Secondly, the maximum permissible pitch angle has been increased to -14 degrees. Finally, it was noticed from the previous flight that the airspeed was used in calculating the roll angle, instead of the ground speed (as required by the MPN algorithm), and this could have resulted in the aircraft flying near the stall speed as stated previously, since the groundspeed is the sum of the airspeed and windspeed. Hence, the GPS speed was taken as the input in calculating the lateral acceleration in

this test.

As seen in Figure 10b, the vertical track error does improve, yet at the cost of the horizontal track error (Figure 10a). Further, when comparing Figures 10c and d, it can also be seen that the roll performance is still much better than that of the pitch, albeit with a noticeable lag in response. This lag can be attributed to the GPS update rate of 4 Hz, and the poor performance in pitch, as well as the noisy input signals, to the reasons discussed in the previous example.

With these results, it is deemed that more testing needs to be conducted with the autopilot to determine the relationship between a given roll and pitch and the actual response, as they do not reflect the results obtained using the HORIZON^{mp} Simulator. The MicroPilot® autopilot was sourced from a third party and has been treated as a closed system in the tests, with only simulated responses used to judged its applicability to the project at hand. It should also be noted that the autopilot PID gains were tuned for the UAV in flight following the recommended settings from the manufacturers, using waypoints that are spaced tens or even hundreds of meters apart. With these waypoints, generally a larger track error is tolerated, which is detrimental to the flight path required to be flown in this research, with waypoints spaced mostly within a meter distance of each other.

After further consideration of the nature of the vertical guidance (MPN) algorithm and the MicroPilot® PID loop structure, it is decided that future testing will be conducted using a two-pronged approach. The first strategy will use the desired altitude to control the aircraft pitch, while the second will use the airspeed to control the aircraft pitch. This is because calculating the pitch from altitude or airspeed is an inherent function of the autopilot, and thus presents a far lower risk than attempting to overwrite the pitch directly, as is the case with the tested MPN algorithm. Future tests will also use an electronic compass calibrated to a greater degree of accuracy as this can affect the accuracy of the lateral path following algorithm.

The main advantage of using the 3-D Dubins path planning algorithm trialled in this research is that the relatively benign manoevres are suitable to all aircraft types. The algorithm is also very fast and can be easily implemented on most Off-the-Shelf hardware and software. In addition, the gentle turns and arcs in the Dubins curves allow an onboard camera enough time to locate suitable landing sites beneath the aircraft. When combined with the MPN guidance algorithm, a smooth path is able to be flown by the aircraft to the desired approach point, thus reducing the risk of unnecessary altitude loss caused by jerky manoevres in an emergency landing. However, as alluded to previously, the gain tuning implemented on the aircraft autopilot must cater for closely spaced waypoints, otherwise, accuracy in following the path will be affected. Another limitation of using the approach discussed is that the target vehicle must be able to receive external roll and pitch commands for the planning and guidance algorithms to work. A final limitation is that the approach discussed is not suitable to a forced landing situation in which the distance between the aimpoint and failure point is so small that a Dubins path cannot be formed.

VI. Conclusion

This paper has presented the results of flight experiments conducted to validate previously developed planning and guidance algorithms for a UAV forced landing. Simulations have demonstrated the validity of the proposed algorithms, and with good results. However, the flight test results are inconsistent with the simulated results, for the reasons given previously, and a solution has been presented which aims to alleviate these incongruencies. It is hoped that by implementing the proposed solution in upcoming flight tests, the performance of the planning and guidance algorithms can be validated.

VII. Acknowledgement

This research was supported under the Australian Research Council DECRA funding scheme (Project No. DE120100802). The authors would like to thank Mr. Richard Glassock and Mr. Scott McNamara for providing technical support during flight trials.

References

[1] US-OSoD, *Unmanned systems integrated roadmap FY2011-2036*. Office of the Secretary of Defense, 2011.

[2] M. T. DeGarmo, "Issues concerning integration of unmanned aerial vehicles in civil airspace," MITRE, Tech. Rep., 2004, mP 04W0000323.

[3] C. R. C. Redelinghuy, "A flight simulation algorithm for a parafoil suspending an air vehicle," *Journal of Guidance Control and Dynamics*, vol. 30, no. 3, pp. 791–803, 2007.

[4] A. Buttler, "Global hawk emergency prompts hard landing," *Aviation Week*, 1 June 2009.

[5] CASA, *Visual flight rules guide (2nd Ed.)*. Camberra: Civil Aviation Safety Authority Australia (CASA), 2007.

[6] L. E. Dubins, "On curves of minimal length with a constraint on average cur- vature with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 471–477, 1957.

[7] S. Kim, P. Silson, A. Tsourdos, and M. Shanmugavel, "Dubins path planning of multiple unmanned airborne vehicles for communication relay," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 1, pp. 12–25, 2011. [Online]. Available: http://pig.sagepub.com/content/225/1/12.abstract

[8] A. R. Babaei and M. Mortazav, "Three-dimensional curvature-constrained trajectory planning based on in-flight waypoints," *Journal of Aircraft*, vol. 47, no. 4, 2010.

[9] P. C. Eng, "Path planning, guidance and control for a uav forced landing," Ph.D. dissertation, Queensland University of Technology, 2011. [Online]. Available: http://eprints.qut.edu.au/43898/

[10] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, E. De Lellis, and A. Pironti, "Path generation and tracking in 3-d for uavs," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 4, pp. 980 –988, july 2009.

[11] P. C. Eng, L. Mejias, R. A. Walker, and D. L. Fitzgerald, "Guided chaos: path planning and control for a uav-forced landing," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 90–98, June 2010. [Online]. Available: http://eprints.qut.edu.au/32800/

[12] S. Park, J. Deyst, and J. P. How, "Performance and lyapunov stability of a nonlinear path-following guidance method," *Journal of Guidance Control and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.

[13] R. Nelson, *Flight stability and automatic control*. McGraw-Hill, 1998.

[14] P. C. Eng, L. Mejias, X. Liu, and R. A. Walker, "Automating human thought processes for a uav forced landing," *Journal of Intelligent and Robotic Systems*, November 2009. [Online]. Available: http://eprints.qut.edu.au/28708/

[15] L. Mejias, D. L. Fitzgerald, P. C. Eng, and L. Xi, "Forced landing technologies for unmanned aerial vehicles : towards safer operations," in *Aerial Vehicles*, L. T. Mung, Ed. Kirchengasse, Austria: In-Tech, January 2009, pp. 413–440. [Online]. Available: http://eprints.qut.edu.au/42556/

[16] P. C. Eng, L. Mejias, R. A. Walker, and D. L. Fitzgerald, "Simulation of a fixed-wing uav forced landing with dynamic path planning," in *Australasian Conference on Robotics and Automation*, Brisbane, Australia, 2007. [Online]. Available: http://eprints.qut.edu.au/11235/

[17] Micropilot, "http://www.micropilot.com," 2012.